Amendments to the Claims:

The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

Claim 1 (original)

Claim 2 (original)

Claim 3 (original)

Claim 4 (currently amended)

Claim 5 (currently amended)

Claim 6 (currently amended)

Claim 7 (original)

Claim 8 (currently amended)

Claim 9 (new)

Claim 10 (new)

Claim 11 (new)

Claim 12 (new)

Claim 13 (new)

Claim 14 (new)

1. (original)

A computer program on a media usable with a computer for testing combinational and sequential logic circuits where memory units are coupled together to form shift register latches that are arranged in a shift register scan path with an input and output for testing the logic circuits, said computer program comprising:

load pattern computer code for shifting data through the scan path to load the shift register latches with a first data pattern representative of a stuck-at fault condition;

pattern variation computer code for causing permutation of at least one of the following operating parameters: a supply voltage, a reference voltage, a timing pattern temperature and a timing sequence to trigger a change in state of at least one of the memory units in the shift register scan path; and

analyzing computer code for determining the memory unit furthest from the shift register scan path output that has changed state from its loaded value.

2. (original)

The computer program of claim 1, wherein said pattern variation computer code is for causing permutations in a plurality of the operating parameters.

3. (original)

The computer program of claim 2, wherein said analyzing computer code includes shifting code for shifting data out of the scan path after each of the operating parameters is separately permuted.

4. (currently amended)

The computer program of claim 3, wherein said analyzing computer code includes selection computer code for selecting the last bit read out that has changed from its load pattern as being from the shift register latch closest to the stuck-at fault ~~memory unit~~ condition.

5. (currently amended)

A method for testing combinational and sequential logic circuits where memory [[;]]units are coupled together to form shift register latches, arranged in a shift register scan path with an input and output for testing the logic circuits, the method comprising:

determining a stuck-at fault condition exists in one of the shift register latches;

shifting data through the scan path to load the shift register latches with a first data pattern representative of [[a]] the outputs state as a result of the stuck-at fault condition;

causing permutation of at least one of the following operating parameters: a supply voltage; a reference voltage; a timing pattern temperature and a timing sequence to trigger a change in state from the stuck-at fault state of at least one of the memory units in the shift register scan path; and

determining the memory unit furthest from the shift register scan path output that has changed state from its loaded value.

6. (currently amended)

The method of claim 5 including:

causing permutations in a plurality of the operating parameters in determining said memory unit furthest from the shift register scan path output.

7. (original)

The method of claim 6 including:

shifting data out of the scan path after each of the operating parameters is separately permuted.

8. (currently amended)

The method of claim 7 including:

selecting the last bit read out that has changed from its load pattern as being from the shift register latch closest to the stuck-at fault memory ~~unit~~ condition.

9. (new)

The method of claim 5 including loading all shift register latches of the scan chain with stuck fault output state.

10. (new)

A computer program on a media usable with a computer for testing combinational and sequential logic circuits where memory units are coupled together to form shift register latches that are arranged in a shift register scan path with an input and output for testing the logic circuits, said computer program comprising:

stuck fault detection code for detecting a stuck-at fault output level of the shift register scan path from an expect state;

load pattern computer code for shifting data through the scan path to load the shift register latches of the scan path with the detected stuck-at fault output level condition;

pattern variation computer code for causing permutation of at least one of the following operating parameters: a supply voltage, a reference voltage, a timing pattern temperature and a timing sequence to trigger a change in state of at least one of the memory units in the shift register scan path that is detectable at the output of the shift register scan path; and

analyzing computer code for determining the memory unit furthest from the shift register scan path output that has changed state from its loaded value as a result of permutations of an operating parameter.

11. (new)

A computer program of claim 10 including masking code for masking out all expects for latches following and including a farthest failing latch.

12. (new)

The computer program of claim ll, wherein said pattern variation computer code is for causing permutations in a plurality of the operating parameters centered around a working threshold varying the operating parameters in the vicinity of the working threshold.

13. (new)

The computer program of claim 12, wherein said analyzing computer code includes shifting code for shifting data out of the scan path after each of the operating parameters is separately permuted.

14. (new)

The computer program of claim 11, wherein said analyzing computer code includes selection computer code for selecting the last bit read out that has changed from its load pattern as being from the shift register latch closest to the stuck-at fault memory unit.